Intelligence Community and Department of Defense

Content Discovery & Retrieval Integrated Project Team

IC-DoD Simple Object Access Protocol (SOAP) Encoding Specification For CDR Query Management

V1_20111214

14 December 2011

i

REVISION/HISTORY

Doc Revision	Revised By	Revision Date	Revisions
V1.0	CDR-IPT	August 2011	Initial release

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Service Overview

The Query Management (QM) Component, as defined by the "IC/DoD Content Discovery and Retrieval (CDR) Specification Framework (SF),"[1] is a CDR Component that manages Saved Searches[2] and may initiate search requests based on Saved Searches.

This specification presents details for the realization of the Content Discovery and Retrieval (CDR) Query Management Component, hereafter termed the Query Management (QM) Service in this document, as a web service using the SOAP style binding. It describes the external service interfaces and the internal activities that provide service behaviors so that service providers and consumers can create and use CDR-compliant Query Management Services.

The Query Management Service provides a coordinated set of functions that enable service consumers to create, read, update, delete, search for, and execute Saved Searches. The resource model presented in Figure 1 provides an overview of the information that supports Query Management functionality. The rectangle in the middle of the figure shows the bundle of information contained within a Saved Search. The Saved Search information includes a Search Request and the target search capability (where the search is to be executed). Each Search Bundle has an associated Saved Search ID that is used to reference the Saved Search Bundle within a QM Collection.

The rectangle on the right-hand-side of Figure 1 emphasizes that the Search Request is consistent with the definition published in the CDR Search Specification. A CDR Search Request consists of the Query that contains the search criteria expressed in a documented format, along with property sets that can be used to provide more information about the query as well as the search itself. The Saved Search Description shown on the left-hand-side of Figure 1 shows the characteristic description metadata that aids in the discovery of Saved Searches.

---

[1] *IC-DoD Content Discovery and Retrieval Specification Framework V1.0.*
[2] *Refer to the 'Common Definitions of Terms related to Search and Query' in the IC-DOD Content Discovery & Retrieval Specification Framework V1.0 for a consistent set of definitions.*
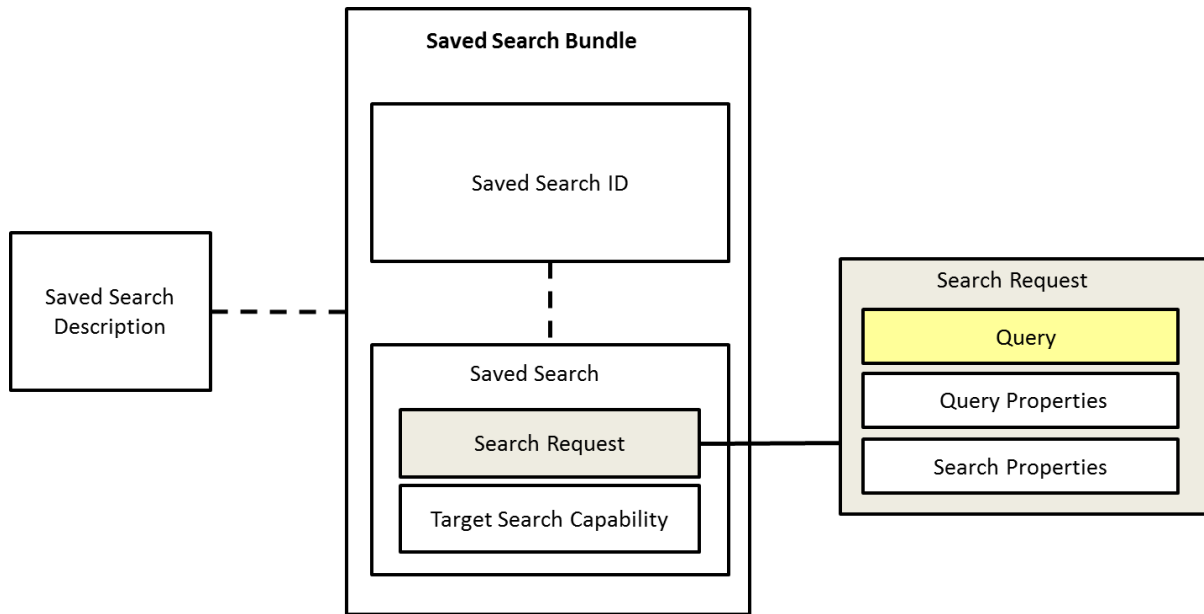
Figure 1. Query Management Resource Model

This specification addresses the representation, management, and use of the Saved Search information bundle. Bundle information is represented within an Atom Entry Document. The Entry serves as the container for the metadata and data associated with a Saved Search. Each 'atom entry' has an associated 'atom entry ID' which is used to reference a Saved Search Bundle and the information it contains. The QM functions for creating, reading, updating, and deleting Saved Search information are based on the protocol operations specified in the Atom Publishing Protocol (AtomPub).3

The ability to save and retrieve queries over time will require implementers to adopt a persistence mechanism, which this document refers to as a QM Collection. Like an Atom Collection, the QM Collection contains a set of resources that can be retrieved in all or in part. The implementation of the QM Collection is not in the scope of this document. This document specifies the standard, implementation-independent interfaces to the functionality provided by the QM Service.

1.2 Relationship to Other CDR Architecture Elements

The CDR Architecture prescribes an abstract-to-concrete model for the development of architecture elements and guidance for CDR. Each layer, or tier, of the model is intended to provide key aspects of the overall guidance to achieve the goals and objectives for joint DoD/IC content discovery and retrieval.

---

3 Atom Publishing Protocol, http://www.ietf.org/rfc/rfc5023.txt

The following graphic in Figure 2, discussed in detail within the CDR Reference Architecture,[4] illustrates this model.
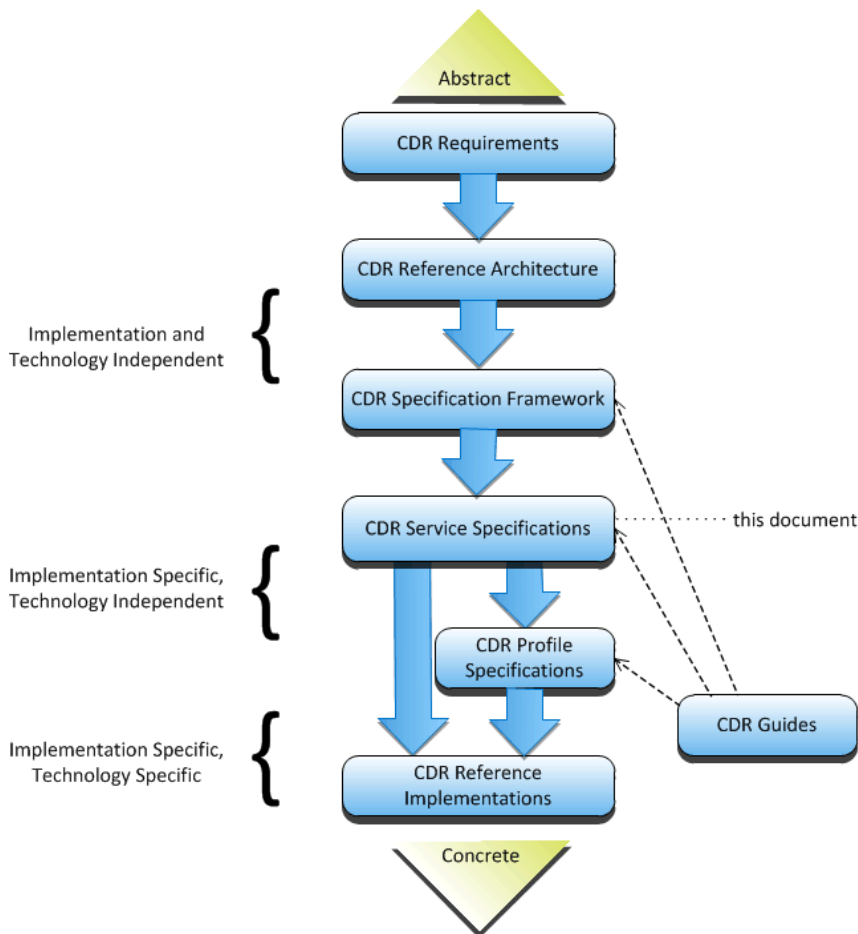
Figure 2. CDR Architectural Model

As illustrated in Figure 2, the CDR Specification Framework derives from the Reference Architecture (RA) and describes behavior in terms of the capabilities, components, and usage patterns defined in the RA. The Specification Framework allows multiple Service Specifications to provide consistent interfaces, both in terms of the structure and semantics of the exchanged information. This specification provides guidance for implementing the CDR Query Management Component using the SOAP service. It is intended to provide minimal requirements for implementing Query Management.

---

[4] *IC/DoD Content Discovery & Retrieval Reference Architecture V1.0*

1.3 Scope

This specification is limited to the interactions that occur between an Initiating Consumer and the Query Management Service as described in the CDR Reference Architecture and CDR Specification Framework.

This specification provides the description of the QM Service Behavior in terms of the message exchange patterns necessary to that enable service consumers to create, read, update, delete, search for, and execute Saved Searches.

1.4 Notational Convention

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as described in the IETF RFC 2119[5]. When these words are not capitalized, they are meant in their natural-language sense.

When describing concrete eXtensible Markup Language (XML) schemas and example XML documents, this specification uses XPath as the notational convention. Each member of an XML schema is described using an XPath notation (e.g., /x:RootElement/x:ChildElement/@Attribute). The use of {any} indicates the presence of an attribute wildcard (<xs:anyAttribute>).

Examples in this text are distinguished by a black border. These are meant to be illustrative and represent one way that the described syntax can be used.

```
<atom:entry>
<atom:title>This is an example.</atom:title>
</atom:entry>
```

Example 1. Example Notation Convention

1.5 Conformance

This specification defines the interface to a Query Management SOAP Service. Query Management implementations conformant with this specification MUST support all of its mandatory terms.

1.6 Namespaces

Table 1 contains namespaces used throughout this document in parameter definitions and examples.

---

[5] Key words for use in RFCs to Indicate Requirement Levels, http://www.ietf.org/rfc/rfc2119.txt.

Table 1. Namespaces

Prefix	URI	Description
soap	http://www.w3.org/2003/05/soap-envelope	W3C SOAP Version 1.2
wsa	http://www.w3.org/2005/08/addressing	WS-Addressing
wsaw	http://www.w3.org/2006/05/addressing/wsdl	WS-Addressing – SOAP Binding
cdrqm	urn:cdr:querymanagement:1.0	CDR Query Management at the indicated version
cdrs	urn:cdr:search:2.0	CDR Search
atom	http://www.w3.org/2005/Atom	Atom Syndication Format

Many of the examples will include an entry such as <atom:entry xmlns…> to indicate that
the full XML would include the appropriate namespace declarations but the full declarations
have not been included as part of the example for brevity and ease of maintaining this
specification.   Any use of namespaces included in Table 1 should be interpreted as defined
in Table 1. The use of elements from the atom namespace is consistent with the Atom
Syndication Format.

2. Service Behavior

QM Service Behavior is implemented through the six fundamental operations, or functions,
that together define service behavior. The six functions listed below provide QM service
consumers with a coordinated set of capabilities that support managing and using both
searches and search related information. The first four functions facilitate the Saved Search
lifecycle that is associated with persistent storage.   This specification mandates the generic
interfaces that a QM consumer would utilize to manage and use a Saved Search.

1. Create -The Create function is used to insert a new Saved Search into the QM
   Collection.
2. Read - The Read function is used to retrieve a Saved Search from the QM Collection.
3. Update -The Update function is used to change a Saved Search in the QM Collection
4. Delete -The Delete function is used to logically remove a Saved Search from the QM
   Collection
5. Execute - The Execute function enables a QM consumer to execute (run) a Saved
   Search at the location specified by the Target Search Capability.   To process an
   execute request, the Query Management Service retrieves the Target Search
   Capability (the location of the Search Service) and the Search Request from the
   Saved Search. This information is then used to initiate the Search. This capability
   SHOULD leverage the CDR Search approach to effect the execution of a Saved
   Search.

6. Search - The Search function enables a prospective consumer to interrogate the QM Collection on the basis of anything to be found in the Saved Search Bundle or the Saved Search Description to determine if a suitable search has already been created. Generic search terms can be used to provide functionality that provides a 'list' of the contents of the QM Collection. This capability SHOULD leverage the CDR Search discovery approach.

The QM functional operations can be implemented using a request/response message pattern. For example, the QM-Create message request supplies information pertinent to the information to be persisted and the QM service will respond with the resource reference and status information.

3. Service Interfaces

This section describes the optional, required, and recommended functions that are specified for the Query Management Service. The description of each function includes:

- A textual description of the function.
- Pre-conditions: A condition that must exist or be established before the service function can be invoked. The preconditions MUST be satisfied in order for the requested QM function is to correctly process input and generate results and post-conditions as described.
- A table of the input and output parameters including whether the parameter is required or optional.
- A textual description of each parameter.
- Post-conditions: The information state after the requested service has successfully executed.
- A SOAP based example.
- A request and a response mapping to the Specification Framework variables from the SOAP Specification elements.

The CDR Query Management functions MUST use SOAP 1.2.[6]

This version of the CDR Query Management Service uses elements, concepts, and constraints from the Atom Syndication Format (Atom).[7] As mentioned in the introduction Atom provides a convenient and applicable format for a Saved Search resource representation. The Atom specification provides the normative guidance for usage of Atom elements within this specification. Where applicable, this QM Specification provides *additional* information and/or constraints necessary for QM implementations. This specification uses the conceptual model for CRUD-like operations as published in the Atom Publishing Protocol (AtomPub) specification.

---

[6] This page (http://www.w3.org/TR/soap) contains links to the SOAP/1.1 note and the SOAP Version 1.2 Recommendation documents.

[7] Query Management Service uses the atom:entry schema as defined in the Atom Syndication Format.

The use of Atom to express Saved Search information in a common payload enables a degree of REST/SOAP interoperability.

The following Table summarizes the service functions defined for Query Management.

<div align="center">

Table 2. Query Management Service Function Summary

Function Name	Requirement Level
QM-Create	REQUIRED
QM-Read	REQUIRED
QM-Update	REQUIRED
QM-Delete	REQUIRED
QM-Execute	REQUIRED
QM-Search	RECOMMENDED

</div>

3.1 QM-Create Function

The QM-Create function enables the user to define and save a search request that can later be accessed and managed through reference to a Saved Search.   In response to a QM-Create request, the QM component implementation will assign a unique resource identifier referred to as a Saved Search ID.

3.1.1 Preconditions

1. A QM collection must exist.

3.1.2 Input

3.1.2.1 SOAP Header

The following are descriptions of the input elements and attributes in the SOAP Header:

/wsa:Action - REQUIRED - This input element indicates the intent of the message. The value MUST be "urn:cdr:querymanagement:1.0:create."

/cdrqm:QMCreateProperties - OPTIONAL - Information provided by the QM consumer to specify and configure optional behavior supported by the QM Create function implementation.

3.1.2.2 SOAP Body

The following are descriptions of the input elements and attributes in the SOAP Body:

/atom:entry[8] - REQUIRED

/atom:entry/atom:id - REQUIRED

QM Recommendation:

- The atom id serves as the Saved Search ID and is used to uniquely identify a Saved Search. The value returned by QM-Create is permanent for the life of a Saved Search resource and is the value for all uses of the Saved Search ID.
- The user may provide a user-defined identifier be used in creating the atom:id, but the QM-Create function implementation is not obligated to honor the user's ID request. Given atom:id is a required input, a value of "urn-defaultID" SHOULD be used if the user does not wish to provide a user-defined identifier.

/atom:entry/atom:title - REQUIRED

/atom:entry/atom:summary - OPTIONAL

/atom:entry/atom:author - REQUIRED

QM Recommendation:
- If available, the name element SHOULD be populated with the name of the organization/program/agency/person authoring the saved search.
- If available, the "uri" element SHOULD refer to a resource providing information about the named author.
- If available, the "email" element SHOULD refer to the POC email for the named author.
-

Example:

```
<atom:author>
    <atom:name>DNI IC SOA Team</atom:name>
    <atom:uri>http://intelink.gov</atom:uri>
    <atom:email>some.address@dni.gov</atom:email>
</atom:author>
```

/atom:entry/atom:updated - REQUIRED

QM Recommendation:

- The initial value of atom:updated SHOULD be the date the Saved Search was created. A cdrqm:SavedSearch MUST contain either a cdrqm:SavedSearchURL OR a cdrs:SearchRequest and cdrqm:TargetSearchCapability.
- The value of atom:updated SHOULD be revised whenever a QM-Update request is successfully processed.

Example:

<div style="border:1px solid black; padding:8px;">
<atom:updated>2003-12-13T18:30:02Z</atom:updated>
</div>

/atom:entry/{SavedSearchDescription} – OPTIONAL – The Saved Search Description is a collection of description elements that are defined by one or more industry, government, or other organizations to further describe a Saved Search.

Example of {SavedSearchDescription}:

<div style="border:1px solid black; padding:8px;">
<pol:AllocationPolicy xmlns:pol = "http://policy.gov">
 JointIC-DoDPolicyDefaults
</pol:AllocationPolicy>
</div>

/atom:entry/atom:content/cdrqm:SavedSearch[9] – REQUIRED - A search request and related information that is managed by implementations of the Query Management Component, and that resides inside of Atom content.

/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL – OPTIONAL (CONDITIONAL) – URL containing all information necessary by a search consumer to initiate a search. The URL location must reference a valid OpenSearch Search Component or Brokered Search Component implementation and the search request MUST be composed of Search Function inputs as defined by the IC/DoD Content Discovery & Retrieval OpenSearch specification.

/atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest – OPTIONAL (CONDITIONAL) - All of the information sent by a search consumer that initiates a search. The search request MUST be composed of Search Function inputs as defined by the IC/DoD SOAP Encoding Specification for Search.

/atom:entry/atom:content/cdrqm:SavedSearch /cdrqm:TargetSearchCapability - OPTIONAL (CONDITIONAL) - Reference to the Search Component or Brokered Search Component implementation that is to process the search request. The reference MUST provide or be able to be transformed to an address through which a Consumer Component can later initiate a search by sending the search request to that address.

---

[9] *The normative definitions for search-related terms in the cdrqm namespace may be found in section 2.3 of the CDR Specification Framework.*

3.1.2.3 Create Request – Message Example

```
<soap:Envelope>
  <soap:Header>
    <wsa:Action>urn:cdrqm:Create</wsa:Action>
    <cdrqm:CreateProperties>
      <cdrqm:ReturnDescription>yes</cdrqm:ReturnDescription>
    </cdrqm:CreateProperties>
  </soap:Header>
  <soap:Body>
    <atom:entry xmlns…>
      <!-- Atom entry elements used by QM -->
      <atom:id>urn:defaultID</atom:id>
      <atom:title>Mohammad Atta Keyword Search</atom:title>
      <atom:summary>search for all occurrences of the named terrorist</atom:summary>
      <atom:author><atom:name>John Smith</atom:name></atom:author>
      <atom:updated>2011-07-13T18:30:02Z</atom:updated>
      <cdrqm:SavedSearchDescription>
        <pol:AllocationPolicy xmlns:pol=http://policy.gov">
          JointIC-DoDPolicyDefaults
        </pol:AllocationPolicy>
      </cdrqm:SavedSearchDescription>
      <atom:content type="application/xml">
        <!-- QM Content -->
        <cdrqm:SavedSearch>
          <cdrs:SearchRequest startIndex="1" count="10"
            queryLanguage="urn:cdr:queryLanguage:lucene" responseFormat="atom+xml">
            <cdrs:Expression>"watson ibm"</cdrs:Expression>
             <cdrs:Sort>created_at desc<cdrs:Sort>
          </cdrs:SearchRequest>
          <cdrqm:TargetSearchCapability>
            <wsa:Address>http://searchservice</wsa:Address>
            <wsa:Metadata>
              <wsaw:InterfaceName>urn:cdr:search:1.0:searchInterface</wsaw:InterfaceName>
            </wsa:Metadata>
          </cdrqm:TargetSearchCapability>
        </cdrqm:SavedSearch>
      </atom:content>
    </atom:entry>
  </soap:Body>
</soap:Envelope>
```

Example 2. Create Request

11

3.1.2.4 Relation to Inputs Defined in the Specification Framework

The IC/DoD CDR Specification Framework defines a number of inputs to the Search operation. The following table relates the disposition of each variable defined in the Framework as it relates to this specification:

Table 3. Specification Framework Create Input Variables Disposition

Specification Framework Variables	SOAP Specification Element
QM Properties	cdrqm:CreateProperties
Search Request	/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL /atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest
Target Search Capability	/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL /atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:author /atom:entry/atom:summary /atom:entry/atom:updated /atom:entry/{SavedSearchDescription}

3.1.3 Output

3.1.3.1 SOAP Header
The following are descriptions of the input elements and attributes in the SOAP Header:

/atom:id - REQUIRED - As defined in Section 3.1.2.2.

QM Recommendation

- The atom:id element conveys a permanent, universally unique identifier for an entry.
- The content of an atom:id element MUST be created in a way that assures uniqueness.
- Instances of atom:id elements can be compared to determine whether an entry or feed is the same as one seen before.

3.1.3.1 SOAP Body

The following are descriptions of the output elements and attributes in the SOAP Body:

/atom:entry - OPTIONAL - As defined in Section 3.1.2.2.

/atom:entry/atom:id - REQUIRED - As defined in Section 3.1.2.2.

Where previously defined, the following atom: and cdrqm: items primarily reflect values as provided in the SOAP request.

/atom:entry/atom:title - REQUIRED - As defined in Section 3.1.2.2.

/atom:entry/atom:summary - OPTIONAL - As defined in Section 3.1.2.2.

/atom:entry/atom:author - REQUIRED - As defined in Section 3.1.2.2.

/atom:entry/atom:updated - REQUIRED - As defined in Section 3.1.2.2.

/atom:entry/{SavedSearchDescription} – OPTIONAL - As defined in Section 3.1.2.2.

/atom:entry/atom:content/cdrqm:SavedSearch - REQUIRED - As defined in Section 3.1.2.2.

/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL - OPTIONAL (CONDITIONAL) - As defined in Section 3.1.2.2.

/atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest - OPTIONAL (CONDITIONAL) - As defined in Section 3.1.2.2.

/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability - OPTIONAL (CONDITIONAL) - As defined in Section 3.1.2.2.

3.1.3.2 Create Response – Message Example

```
<soap:Envelope>
  <soap:Body>
    <atom:entry xmlns…>
      <!-- Atom entry elements used by QM -->
      <atom:id> urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1 </atom:id>
      <atom:title> Mohammad Atta Keyword Search </atom:title>
      <atom:author><atom:name>John Smith</atom:name></atom:author>
      <atom:summary> search for all occurrences of the named terrorist </atom:summary>
      <atom:updated> 2011-07-13T18:30:02Z </atom:updated>
      <atom:content type="application/xml">
        <cdrqm:SavedSearch>
          <cdrs:SearchRequest xmlns:s="Search Component">
            <cdrs:Query>sample query</cdrs:Query>
            <cdrs:QueryProperties>
               <QueryLanguage>Xquery</QueryLanguage>
            </cdrs:QueryProperties>
            <cdrs:SearchProperties>
               <cdrs:timeout>200</cdrs:timeout>
               <cdrs:resultsPerPage>100</cdrs:resultsPerPage>
             </cdrs:SearchProperties>
          </cdrs:SearchRequest>
          <cdrqm:TargetSearchCapability>
            <wsa:Address>http://searchservice</wsa:Address>
            <wsa:Metadata>
               <wsaw:InterfaceName>cdrs:SearchInterface</wsaw:InterfaceName>
            </wsa:Metadata>
          </cdrqm:TargetSearchCapability>
        </cdrqm:SavedSearch>
      </atom:content>
    </atom:entry>
  </soap:Body>
</soap:Envelope>
```

Example 3. Create Response

3.1.3.3 Post-condition

1. The Saved Search Resource is available for QM-Read, QM-Update, QM-Delete, QM-Search, and QM-Execute and it is identifiable by the Saved Search ID.
2. The Create function has been audited according to applicable policy.[10]

---

[10] The Create function may be audited according to applicable policy regardless to the success or failure of the function.

3.1.3.4 Relation to Outputs Defined in the Specification Framework

Table 4. Specification Framework Create Output Variables Disposition

Specification Framework Variables	SOAP Specification Element
Saved Search ID	/atom:id /atom:entry/atom:id
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:author /atom:entry/atom:summary /atom:entry/atom:updated /atom:entry/cdrqm:SavedSearchDescription

3.2 QM-Read Function

The QM-Read function uses the Saved Search ID to get/retrieve a Saved Search Bundle from the QM Collection.   Since there is a one-to-one correspondence between the Saved Search Bundle (expressed as an atom:entry) and a Saved Search (encapsulated in the atom:content element) the unique atom:id that is associated with the bundle uniquely identifies a Saved Search.

3.2.1 Preconditions

1. Saved Search is under management of QM and may be retrieved through reference to its Saved Search ID.

3.2.2 Input

3.2.2.1 SOAP Header

The following are a description of the input elements and attributes in the SOAP Header:

/wsa:Action - REQUIRED - This input element indicates the intent of the message. The value MUST be "urn:cdr:querymanagement:1.0:read."

/atom:id - REQUIRED - As defined in Section 3.1.2.2.

/cdrqm:QMReadProperties - OPTIONAL - Information provided by the QM consumer to specify and configure optional behavior supported by the QM Read Component implementation.

15

3.2.2.3 Read Request – Message Example

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  <atom:entry xmlns…>
   <soap:Header>
      <wsa:Action>Retrieve</wsa:Action>
      <atom:id> urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1 </atom:id>
      <cdrqm:ReadProperties>
        ...
      </cdrqm:ReadProperties>
   </soap:Header>
   <soap:Body/>
 </soap:Envelope>
```

Example 4. Read Request

3.2.2.4 Relation to Read Inputs Defined in the Specification Framework

Table 5. Specification Framework Read Input Variables Disposition

Specification Framework Variables	SOAP Specification Element
Saved Search ID	/atom:id
QM Properties	/cdrqm:QMReadProperties

3.2.3 Output

3.2.3.1 SOAP Body

The following are descriptions of the output elements and attributes in the SOAP Body :

/atom:entry - REQUIRED - As defined in Section 3.1.2.2.

3.2.3.2 Read Response – Message Example

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
   xmlns:cdrs="urn:cdr:search:1.0">
   <soap:Body>
     <atom:entry xmlns…>
        <atom:id> urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1 </atom:id>
        <atom:title> Mohammad Atta Keyword Search </atom:title>
        <atom:author><atom:name>John Smith</atom:name></atom:author>
        <atom:summary> search for all occurrences of the named terrorist </atom:summary>
        <atom:updated> 2011-07-13T18:30:02Z </atom:updated>
        <atom:content type="application/xml">
          <cdrqm:SavedSearch>
            <cdrs:SearchRequest startIndex="1" count="10"
               queryLanguage="urn:cdr:queryLanguage:lucene" responseFormat="atom+xml">
              <cdrs:Expression>"watson ibm"</cdrs:Expression>
              <cdrs:Sort>created_at desc<cdrs:Sort>
            </cdrs:SearchRequest>
            <cdrqm:TargetSearchCapability>
              <wsa:Address>http://searchservice</wsa:Address>
              <wsa:Metadata>
                 <wsaw:InterfaceName>cdrs:SearchInterface</wsaw:InterfaceName>
              </wsa:Metadata>
            </cdrqm:TargetSearchCapability>
          </cdrqm:SavedSearch>
        </atom:content>
        <!-- should add links (endpoint references to retrieve, edit, and delete -->
     </atom:entry>
   </soap:Body>
</soap:Envelope>
```

Example 5. Read Response

3.2.3.3 Post-conditions

1. Saved Search Bundle is not affected by read.
2. The Read function has been audited according to applicable policy.[11]

3.2.3.4 Relation to Read Inputs Defined in the Specification Framework

Table 6. Specification Framework Read Output Variables Disposition

Specification Framework Variables	SOAP Specification Element
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:author /atom:entry/atom:summary /atom:entry/atom:updated /atom:entry/{SavedSearchDescription}

3.3 QM-Update Function

The QM-Update function allows a Consumer Component to change a Saved Search. The Saved Search ID uniquely identifies the Saved Search to be modified. Partial updates are not allowed; therefore the QM-Update request MUST send a complete resource representation that is used to replace the corresponding Saved Search. The Saved Search ID will remain the same. It MAY be necessary to retrieve the Saved Search prior to performing the update.

3.3 1 Preconditions

1. Saved Search is under management of QM and may be accessed through reference to its Saved Search ID for purposes of update.

3.3.2 Input

3.3.2.1 SOAP Header

The following are a description of the input elements and attributes in the SOAP Header:

/wsa:Action - REQUIRED -This input element indicates the intent of the message. The value MUST be "urn:cdr:querymanagement:1.0:update."

---

[11] The Read function may be audited according to applicable policy regardless to the success or failure of the function.

/atom:id - REQUIRED - As defined in Section 3.1.2.2.

/cdrqm:QMUpdateProperties - OPTIONAL - Information provided by the QM consumer to specify and configure optional behavior supported by the QM Update function implementation.

3.3.2.2 SOAP Body

The following are a description of the input elements and attributes in the SOAP Body:

/atom:entry - REQUIRED - As defined in Section 3.1.2.2.

3.3.2.3 Update Request – Message Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  <soap:Header>
    <wsa:Action>Update</wsa:Action>
    <atom:id> urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1 </atom:id>
    <cdrqm:UpdateProperties>
      <cdrqm:ReturnDescription>yes</cdrqm:ReturnDescription>
    </cdrqm:UpdateProperties>
  </soap:Header>
  <soap:Body>
    <atom:entry xmlns…>
      <atom:id>Sample URI (should be same as header element)</atom:id>
      <atom:title> Mohammad Atta Keyword Search </atom:title>
      <atom:summary> search for all occurrences of the named terrorist </atom:summary>
      <atom:updated> 2011-07-13T18:30:02Z </atom:updated>
      <atom:content type="application/xml">
        <cdrqm:SavedSearch>
          <cdrqm:TargetSearchCapability>
            <wsa:Address>http://searchservice</wsa:Address>
            <wsa:Metadata>
              <wsaw:InterfaceName>cdrs:SearchInterface</wsaw:InterfaceName>
            </wsa:Metadata>
          </cdrqm:TargetSearchCapability>
          <cdrs:SearchRequest>
            <cdrs:Query>sample query</cdrs:Query>
            <cdrs:QueryProperties>
              <QueryLanguage>Xquery</QueryLanguage>
            </cdrs:QueryProperties>
            <cdrs:SearchProperties>
              <cdrs:timeout>200</cdrs:timeout>
              <cdrs:resultsPerPage>100</cdrs:resultsPerPage>
            </cdrs:SearchProperties>
          </cdrs:SearchRequest>
        </cdrqm:SavedSearch>
      </atom:content>
    </atom:entry>
  </soap:Body>
</soap:Envelope>
```

Example 6. Update Request

21

3.3.2.4 Relation to Update Inputs Defined in the Specification Framework

Table 7. **Specification Framework Update Input Variables Disposition**

Specification Framework Variables	SOAP Specification Element
Saved Search ID	/atom:id /atom:entry/atom:id
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
QM Properties	/cdrqm:QMUpdateProperties

3.3.3 Output

3.3.3.1 SOAP Body

The following are descriptions of the output elements and attributes in the SOAP Body :

/atom:entry - REQUIRED - As defined in Section 3.1.3.1.

3.3.3.2 Update Response – Message Example

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  <soap:Body>
    <atom:entry xmlns…>
      <atom:id> urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1 </atom:id>
      <atom:title> Mohammad Atta Keyword Search </atom:title>
      <atom:author><atom:name>John Smith</atom:name></atom:author>
      <atom:summary> search for all occurrences of the named terrorist </atom:summary>
      <atom:updated>2011-07-13T18:30:02Z </atom:updated>
      <atom:content type="application/xml">
        <cdrqm:SavedSearch>
            <cdrs:SearchRequest startIndex="1" count="10"
               queryLanguage="urn:cdr:queryLanguage:lucene"            \
               responseFormat="atom+xml">
          <cdrs:Expression>"watson ibm"</cdrs:Expression>
         <cdrs:Sort>created_at desc<cdrs:Sort>
          </cdrs:SearchRequest>
          <cdrqm:TargetSearchCapability>
            <wsa:Address>http://searchservice</wsa:Address>
            <wsa:Metadata>
               <wsaw:InterfaceName>cdrs:SearchInterface</wsaw:InterfaceName>
            </wsa:Metadata>
          </cdrqm:TargetSearchCapability>
        </cdrqm:SavedSearch>
      </atom:content>
    </soap:Body>
</soap:Envelope>
```

Example 7. Update Response

3.3.3.3 Post-conditions

1. Saved Search Bundle reflects specified updates.
2. Saved Search Bundle is referenced by the Saved Search ID.
3. The Update function has been audited according to applicable policy.[12]

3.3.3.4 Relation to Update Inputs Defined in the Specification Framework

Table 8. Specification Framework Update Output Variables Disposition

Specification Framework Variables	SOAP Specification Element
Saved Search ID	/atom:entry/atom:id
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:author /atom:entry/atom:summary /atom:entry/atom:updated /atom:entry /{SavedSearchDescription}

3. 4 QM-Delete Function

The QM-Delete function removes a Saved Search resource from the Saved Search collection managed by the QM Component. The CDR Specification Framework includes a discussion of the design considerations related to the QM-Delete function.

3.4.1 Preconditions

1. Saved Search is under management of QM and may be accessed through reference to its Saved Search ID for purposes of delete.

3.4.2 Input

3.4.2.1 SOAP Header

The following are descriptions of the input elements and attributes in the SOAP Header:

/wsa:Action - REQUIRED - This input element indicates the intent of the message. The value MUST be "urn:cdr:querymanagement:1.0:delete."

---

[12] The Update function may be audited according to applicable policy regardless to the success or failure of the function.

/atom:id - REQUIRED - As defined in Section 3.1.2.2.

/cdrqm:DeleteProperties - OPTIONAL - Information provided by the QM consumer to specify and configure optional behavior supported by the QM Delete Component implementation.

3.4.2.2 Delete Request – Message Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  <soap:Header>
    <wsa:Action>Delete</wsa:Action>
    <atom:id>  urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1 </atom:id>
    <cdrqm:DeleteProperties>
      ...
    </cdrqm:DeleteProperties>
  </soap:Header>
  <soap:Body/>
</soap:Envelope>
```

Example 8. Delete Request

3.4.2.3 Relation to Delete Inputs Defined in the Specification Framework

Table 9. Specification Framework Delete Input Variables Disposition

Specification Framework Variables	SOAP Specification Element
Saved Search ID	/atom:id
QM Properties	/cdrqm:QMDeleteProperties

3.4.2.5 Post-conditions

1. Saved Search Bundle is no longer accessible by QM functions.
2. The Delete function has been audited according to applicable policy.[13]

3.4.3 Delete Outputs

3.4.3.1 SOAP Body

A successful QM-Delete function will not return a response.   An unsuccessful QM-Delete function will return the appropriate fault.

Table 10. Specification Framework Delete Output Variables Disposition

---

[13] The Delete function may be audited according to applicable policy regardless to the success or failure of the function.

Specification Framework Variables	SOAP Specification Element
Confirmation	Unsuccessful QM-Delete function will return the appropriate fault

3. 5 QM-Execute Function

The QM-Execute function leverages a CDR Search to submit a Search Request to a specified location, where both the Search Request and the location are contained within a Saved Search that is managed in a QM Collection. In particular, QM-Execute uses a modified CDR Search Interface as specified in the "IC-DoD Content Discovery & Retrieval Search Service Specification for OpenSearch Implementations 1.1", with the Query being replaced by a Saved Search ID. When the request message is received by the Query Management Service, it uses the Saved Search ID to retrieve the corresponding Saved Search from the QM Collection and sends the Search Request to the location specified in the query or in the Target Search Capability. In accordance with the CDR Search Specification, the Search Service builds a set of items, called Search Results, which match the Query.   The Search Results support HTML and Atom response formats. Search Properties specified in the QM-Execute request supersede the property values contained in the Saved Search.

3.5.1 Preconditions

1. Saved Search is under management of QM and may be retrieved through reference to its Saved Search ID.

3.5.2 Input

3.5.2.1 SOAP Header

The following are descriptions of the input elements and attributes in the SOAP Header:

/wsa:Action - REQUIRED - This input element indicates the intent of the message. The value MUST be "urn:cdr:querymanagement:1.0:execute."

/atom:id - REQUIRED - As defined in Section 3.1.2.2.

/cdrqm:QMSearchProperties -
OPTIONAL - Information provided by the QM consumer to specify and configure optional behavior supported by the referenced Search Component or Brokered Search Component.

/cdrqm:QMExecuteProperties - OPTIONAL - Information provided by the QM consumer to specify and configure optional behavior supported by the QM Execute Component implementation.

3.5.2.2 Execute Request – Message

```
<soap:Envelope>
   <soap:Header>
      <wsa:Action>urn:cdrqm:Execute</wsa:Action>
      <atom:id> urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1 </atom:id>
      <cdrqm:ExecuteProperties>
           ...
      </cdrqm:ExecuteProperties>
   </soap:Header>
   <soap:Body/>
</soap:Envelope>
```

Example 9. Execute Request

3.5.2.3 Relation to Execute Inputs Defined in the Specification Framework

Table 11. Specification Framework Execute Input Variables Disposition

Specification Framework Variables	SOAP Specification Element
Saved Search ID	/atom:entry/atom:id
Modified Search Function Inputs	/cdrs:SearchProperties
QM Properties	/cdrqm:QMExecuteProperties

3.5.3 Execute Output

The Output and Post-conditions are those as specified in the "IC-DoD Content Discovery & Retrieval Search Service Specification for SOAP Implementations 2.0" Search function.

3.6 QM-Search Function

The QM-Search function provides the capability of listing or searching the QM Collection, which is the repository of Saved Searches.   QM-Search MUST be compliant with CDR Search Interface as Specified in the "IC-DoD Content Discovery & Retrieval Search Service Specification for SOAP Implementations 2.0". As described in the CDR Search specification, a SOAP Query (including corresponding search attributes) is sent to a content collection's SOAP compliant service. The Search Service builds a set of items, called Search Results, which match the Query.

The Preconditions, Inputs, Output and Post-conditions are those as specified in the "IC-DoD Content Discovery & Retrieval Search Service Specification for SOAP Implementations 2.0" Search function.

27

4. General Fault Conditions

4.1 Fault Handling in SOAP

Different versions of SOAP may have different fault handling syntaxes. Deliver Services MUST use the primary fault handling mechanism for the version of SOAP they support and to which the service is bound. In the following example, an Unsupported Identifier Type fault is returned using the SOAP 1.2 syntax:

Example:

```
<soap:Fault>
 <soap:Code>
  <soap:Value>soap:Sender</soap:Value>
 </soap:Code>
 <soap:Reason>
  <soap:Text>Identifier Execution Fault</soap:Text>
  <soap:Text>Service could not retrieve the specified resource</soap:Text>
 </soap:Reason>
</soap:Fault>
```

Example 10. Example SOAP Fault

5. Web Service Description Language

The Web Service Description Language (WSDL) document that specifies the bindings for the Deliver Service is implementation specific. A WSDL template that provides an initial reference for WSDL development is provided as a supplement to this specification.

6. References

This section includes additional references that may be used to provide further insight into the overall de sign concepts that serve to guide the CDR-IPT engineering efforts.

6.1 Content Discovery and Retrieval References

The CDR Reference Architecture and Specification Framework provide essential background and context to service designers. This document was based on the following CDR Reference Architecture and Specification Framework document versions:

- "IC/DoD Content Discovery and Retrieval Reference Architecture Version 1.1", 25 February 2011.

- "IC/DoD Content Discovery and Retrieval Specification Framework Version 1.0, 9 May 2011.
- IC/DoD SOAP Interface Encoding Specification for CDR Search, DRAFT, Version 2.0", 14 December 2011.

The most recent version of the documents can be found at the unclassified Intelink web site in the Information Access and Discovery focus area section.